IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for managing memory, comprising:

maintaining a memory pool [[;]] that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory, and maintaining the memory pool includes:

assigning to each of the plurality of processes a corresponding amount of committed memory from the memory pool;

specifying a specified amount of memory in the memory pool for allocation for a memory-requesting process;

requesting a memory-releasing process to release a requested amount of memory in the memory pool;

wherein the memory-requesting process is not necessarily the same process as the memory-releasing process.

2. (Previously Presented) A method for managing memory as recited in Claim 1, wherein the memory-releasing process operates in a garbage-collected environment.

3. (Previously Presented) A method for managing memory as recited in Claim 1, wherein the memory-releasing process is a Java process.

4. (Previously Presented) A method for managing memory as recited in Claim 1, wherein the memory-releasing process is a Java program.

5. (Cancelled)

6. (Original) A method for managing memory as recited in Claim 1, wherein the memory pool includes reserved memory.

7. (Currently Amended) A method for managing memory as recited in Claim 1, wherein the memory pool includes memory owned by [[a]] the plurality of processes.

8. (Original) A method for managing memory as recited in Claim 1, wherein the memory pool includes a plurality of subpools.

9. (Original) A method for managing memory as recited in Claim 1, further comprising determining that the specified amount of memory is required for allocation.

10. (Previously Presented) A method for managing memory as recited in Claim 1, further comprising receiving a memory request from the memory-requesting process for the specified amount of memory and determining that the specified amount of memory is required for allocation.

11. (Previously Presented) A method for managing memory as recited in Claim 1, further comprising monitoring a monitored process and determining that the monitored process requires additional memory.

12. (Previously Presented) A method for managing memory as recited in Claim 1, further comprising monitoring a monitored process and detecting a rate of garbage collection for the monitored process.

13. (Previously Presented) A method for managing memory as recited in Claim 1, further comprising selecting the memory-releasing process from a plurality of processes based on status information.

14. (Previously Presented) A method for managing memory as recited in Claim 1, wherein requesting the memory-releasing process to release a requested amount of memory in the memory pool includes making a request via a system call.

15. (Previously Presented) A method for managing memory as recited in Claim 1, wherein requesting the memory-releasing process to release a requested amount of memory in the memory pool includes making a request via an inter-process communication protocol.

16. (Original) A method for managing memory as recited in Claim 1, wherein the specified amount of memory is approximately equal to the requested amount of memory.

17. (Original) A method for managing memory as recited in Claim 1, wherein the requested amount of memory in the memory pool is freeable memory.

18. (Previously Presented) A method for managing memory as recited in Claim 1, further comprising refilling a subpool of the memory pool with the requested amount of memory released by the memory-releasing process.

19. (Currently Amended) A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

maintaining a memory pool [[;]] that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory, and maintaining the memory pool includes:

assigning to each of the plurality of processes a corresponding amount of committed memory from the memory pool;

specifying a specified amount of memory in the memory pool for allocation for a memory-requesting process;

requesting a memory-releasing process to release a requested amount of memory in the memory pool;

wherein the memory-requesting process is not necessarily the same process as the memory-releasing process.

20. (Currently Amended) A memory management system, comprising:

a memory including a memory pool that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory;

a processor coupled to the memory ~~pool~~, configured to:

~~maintain the memory pool;~~

assign to each of the plurality of processes a corresponding amount of committed memory from the memory pool;

specify a specified amount of memory in the memory pool for allocation for a memory-requesting process;

request a memory-releasing process to release a requested amount of memory in the memory pool;

wherein the memory-requesting process is not necessarily the same process as the memory-releasing process.

21. (Currently Amended) A method for managing memory, comprising:

maintaining a memory pool [[;]] that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory, and maintaining the memory pool includes:

assigning to each of the plurality of processes a corresponding amount of committed memory from the memory pool;

receiving status information from [[a]] the plurality of processes; and

managing memory among the plurality of processes using the status information.

22. (Original) A method for managing memory as recited in Claim 21, wherein managing memory among the plurality of processes includes allocating memory to one of the plurality of processes.

23. (Original) A method for managing memory as recited in Claim 21, wherein managing memory among the plurality of processes includes requesting one of the plurality of processes to release memory.

24. (Original) A method for managing memory as recited in Claim 21, wherein the status information includes status of freeable memory.

25. (Original) A method for managing memory as recited in Claim 21, wherein the status information includes efficiency of the process's garbage collector.

26. (Original)A method for managing memory as recited in Claim 21, further comprising requesting status information.

27. (Original) A method for managing memory as recited in Claim 21, wherein the status information is sent along with a memory allocation request.

28. (Original) A method for managing memory as recited in Claim 21, wherein the status information is received periodically.

29. (Currently Amended) A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

maintaining a memory pool [[;]] that is shared by a plurality of processes, wherein the memory pool includes a committed portion and an uncommitted portion, and maintaining the memory pool includes:

assigning to each of the plurality of processes corresponding committed memory from the committed portion, the committed memory being memory required to run said each process;

receiving status information from [[a]] the plurality of processes; and

managing memory among the plurality of processes using the status information.

30. (Currently Amended) A memory management system, comprising:

a memory including a memory pool that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory;

a processor coupled to the memory ~~pool~~, configured to:

~~maintain a memory pool;~~

assign to each of the plurality of processes a corresponding amount of committed memory from the committed portion, the committed memory being memory required to run said each process;

receive status information from a plurality of processes; and

manage memory among the plurality of processes using the status information.

31. (Currently Amended) A method for managing memory, comprising:

determining an appropriate amount of committed memory in a memory pool that is shared by a plurality of processes, wherein the memory pool includes committed memory and uncommitted memory;

maintaining the memory pool;

determining that an amount of uncommitted memory in the memory pool is required for allocation; and

allocating the required amount of uncommitted memory from ~~an uncommitted portion of~~ the memory pool to a process.

32. (Original) A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for allocation includes determining a request priority.

33. (Original) A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for allocation includes receiving a request having an urgency level and determining a request priority based on the urgency level.

34. (Original) A method for managing memory as recited in Claim 31, wherein determining that an amount of memory in the memory pool is required for allocation includes determining whether the process includes memory collateral.

35. (Original) A method for managing memory as recited in Claim 31, wherein allocating the amount of memory to a process includes selecting the amount of memory from a subpool of the memory pool.

36. (Original) A method for managing memory as recited in Claim 31, wherein allocating the amount of memory to a process includes selecting the amount of memory from a subpool of the memory pool; further comprising refilling the subpool with released memory by the process.

37. (Currently Amended) A computer program product for managing memory, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

determining an appropriate amount of committed memory in a memory pool, wherein the memory pool is shared by a plurality of processes, and the memory pool includes committed memory and uncommitted memory;

maintaining the memory pool;

determining that an amount of uncommitted memory in the memory pool is required for allocation; and

allocating the required amount of uncommitted memory from ~~an uncommitted portion of~~ the memory pool to a process.

38. (Currently Amended) A memory management system, comprising:

a memory including a memory pool, wherein the memory pool is shared by a plurality of processes, and the memory pool includes committed memory and uncommitted memory;

a processor coupled to the memory ~~pool~~, configured to:

determine an appropriate amount of committed memory in [[a]] the memory pool;

maintain the memory pool;

determine that an amount of uncommitted memory in the memory pool is required for allocation; and

allocate the required amount of uncommitted memory from ~~an uncommitted portion of~~ the memory pool to a process.